FOP Lab Assignment

NAME: Muhammad Bin Ahsan

CMS: 468098

SECTION: A

TASK 1:

#include <iostream>

using namespace std;

int main() {

// Declare a variable to store the user input

int num;

// Ask the user to enter a number

cout << "Enter an integer (positive) "<<endl;;

cin >> num;

// Display factors using a for loop

cout << "Factors of " << num << " are "<<endl;

for (int i = 1; i <= num; ++i) {

// Check if 'i' is a factor of 'number'

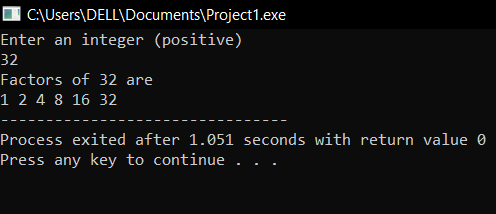
if (num % i == 0) {

cout << i << " ";

}

}

}



TASK 3:

#include <iostream>

using namespace std;

int main() {

// Declare a variable to store the user input

int num;

// Ask the user to enter an integer

cout << "Enter an integer"<<endl;

cin >> num;

// Check if the number is greater than 10 and less than or equal to 20

if (num > 10 && num <= 20) {

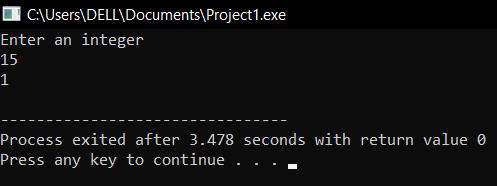
cout << "1" <<endl;

} else {

cout << "0" << endl;

}

}



TASK 4:

#include <iostream>

using namespace std;

// Function to check if a number is prime

bool isPrime(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i \* i <= num; ++i) {

if (num % i == 0) {

return false;

}

}

return true;

}

int main() {

// Input from the user

int N;

cout << "Enter a positive integer N"<<endl;

cin >> N;

// Find the largest prime number less than or equal to N using a while loop

int largestPrime = N;

while (largestPrime > 1) {

if (isPrime(largestPrime)) {

break;

}

--largestPrime;

}

// Output the result

if (largestPrime > 1) {

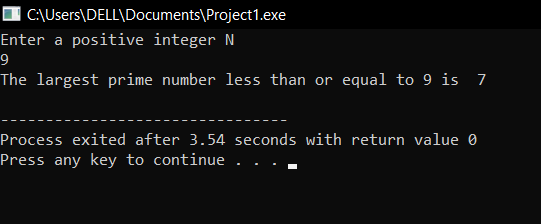
cout << "The largest prime number less than or equal to " << N << " is " << largestPrime << endl;

} else {

cout << "No prime numbers found less than or equal to " << N <<endl;

}

}



TASK 5:

#include <iostream>

#include <algorithm>

using namespace std;

int main() {

// Input two strings from the user

string str1, str2;

cout << "Enter the first string"<<endl;

cin >> str1;

cout << "Enter the second string"<<endl;

cin >> str2;

// Check if the strings are equal

if (str1 == str2) {

// Rotate one of the strings to make them unequal

rotate(str1.begin(), str1.begin() + 1, str1.end());

// Output the modified strings

cout << "After rotation, the first string is: " << str1 <<endl;

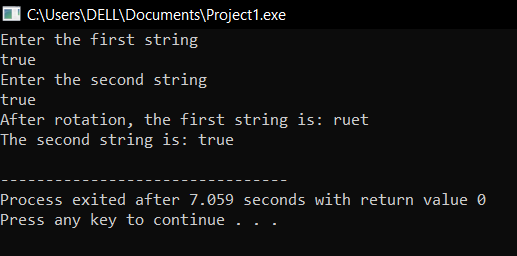
cout << "The second string is: " << str2 << endl;

} else {

cout << "The entered strings are already unequal." << endl;

}

}



TASK 6:

#include <iostream>

using namespace std;

int main() {

// Input dividend and divisor

int dividend, divisor;

cout << "Enter the dividend"<<endl;

cin >> dividend;

cout << "Enter the divisor"<<endl;

cin >> divisor;

// Check if the divisor is zero

if (divisor == 0) {

cout << "Error: Division by zero is undefined." <<endl;

return 1;

}

// Perform division without using '/'

int quotient = 0;

while (dividend >= divisor) {

dividend -= divisor;

quotient++;

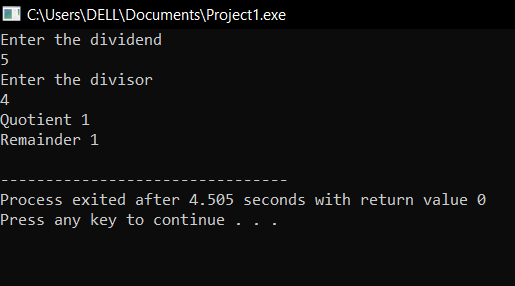
}

// Display the results using '/'

cout << "Quotient " << quotient << endl;

cout << "Remainder " << dividend <<endl;

}



TASK 8:

#include <iostream>

using namespace std;

int main() {

// Original array

int a[5] = {1, 2, 3, 4, 5};

// New size for the array

const int newSize = 8;

// Create a new array with a larger size

int newArray[newSize] = {0};

// Copy elements from the original array to the new array

for (int i = 0; i < 5; ++i) {

newArray[i] = a[i];

}

// Add more elements to the new array

for (int i = 5; i < newSize; ++i) {

newArray[i] = i + 1; // Adding elements 6, 7, 8 to the array

}

// Display the elements in the new array

cout << "Elements in the new array"<<endl;

for (int i = 0; i < newSize; ++i) {

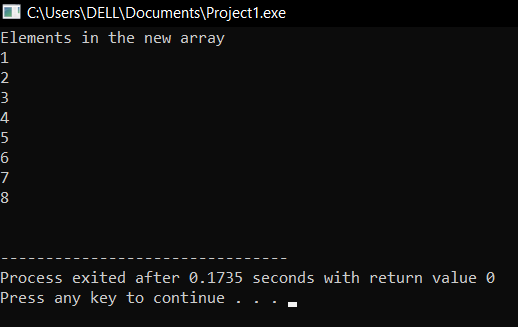
cout << newArray[i] << " "<<endl;

}

cout << endl;

return 0;

}



TASK 9:

#include <iostream>

#include <algorithm>

using namespace std;

bool findTriplet(int arr[], int n, int targetSum) {

// Sort the array

sort(arr, arr + n);

// Fix the first element and use two pointers approach for the remaining elements

for (int i = 0; i < n - 2; ++i) {

int left = i + 1;

int right = n - 1;

while (left < right) {

int currentSum = arr[i] + arr[left] + arr[right];

if (currentSum == targetSum) {

// Triplet found

cout << "Triplet found: " << arr[i] << ", " << arr[left] << ", " << arr[right] << endl;

return true;

} else if (currentSum < targetSum) {

// Move the left pointer to increase the sum

++left;

} else {

// Move the right pointer to decrease the sum

--right;

}

}

}

// No triplet found

return false;

}

int main() {

// Example usage

int arr[] = {1, 4, 45, 6, 10, 8};

int n = sizeof(arr) / sizeof(arr[0]);

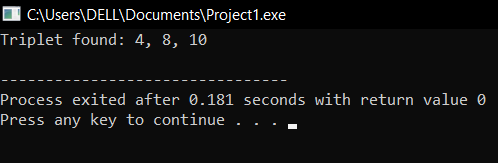
int targetSum = 22;

if (!findTriplet(arr, n, targetSum)) {

cout << "No triplet found with the given sum." << endl;

}

}



TASK 10:

#include <iostream>

using namespace std;

void bubbleSort(int arr[], int size) {

for (int i = 0; i < size - 1; ++i) {

for (int j = 0; j < size - i - 1; ++j) {

// Swap if the element found is greater than the next element

if (arr[j] > arr[j + 1]) {

swap(arr[j], arr[j + 1]);

}

}

}

}

int main() {

const int size = 6;

int arr[size];

// Input array elements

cout << "Enter " << size << " integers for the array" <<endl;

for (int i = 0; i < size; ++i) {

cin >> arr[i];

}

// Perform Bubble Sort

bubbleSort(arr, size);

// Display the sorted array

cout << "Sorted array using Bubble Sort"<<endl;

for (int i = 0; i < size; ++i) {

cout << arr[i] << " "<<endl;

}

cout << endl;

return 0;

}

